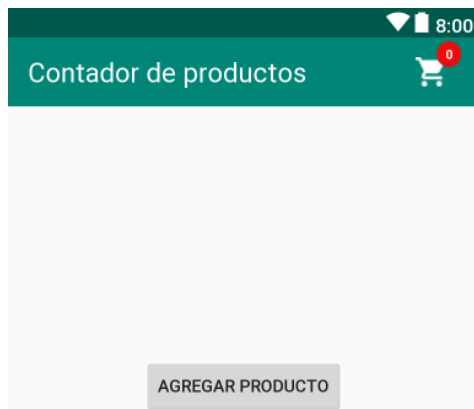


Vamos a implementar una aplicación que incluirá en su ToolBar un carrito de la compra con un contador de los elementos que se han incorporado en el mismo. El aspecto de la activity principal es el siguiente:



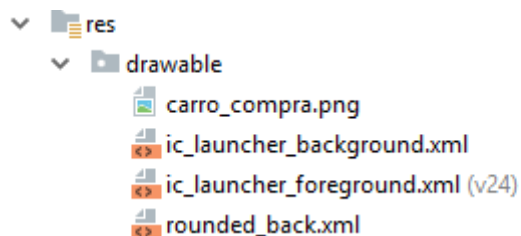
En la activity incluiremos un botón que será el encargado de simular una nueva compra.

Evidentemente es un ejemplo simulado de compra, porque no tenemos ni productos, ni gestionamos qué hacer posteriormente con el carrito. Solamente pretendemos visualizar el diseño y gestión del carrito y del contador incorporado al mismo.

## PASO1

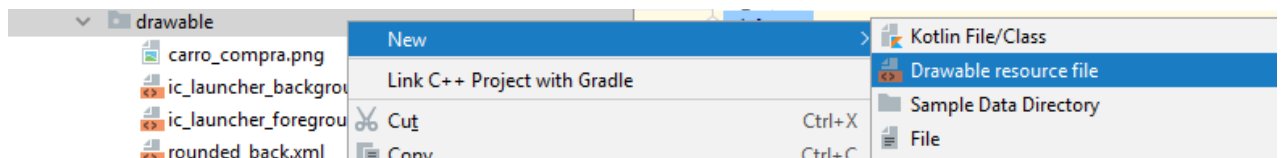
Incluir la librería de diseño a nuestro proyecto.

Incluir los recursos de imagen en la carpeta de resources/drawable:



El archivo carro\_compra.png o se descarga de la página de recursos de iconos de MaterialDesign o estará en la carpeta de recursos de FTP.

Sin embargo el archivo rounded\_back.xml, lo vamos a definir nosotros como un recurso propio. Para ello pulsamos el botón derecho sobre la carpeta drawable y seleccionamos la opción que aparece en la imagen:



Introducimos el nombre del archivo rounded\_back y definimos su contenido:

```
activity_main.xml x rounded_back.xml x MainActivity.java x
1 <?xml version="1.0" encoding="utf-8"?>
2 <shape xmlns:android="http://schemas.android.com/apk/res/android"
3     android:shape="oval">
4     <solid android:color="#ea0d0d"/>
5     <size android:height="20dp"
6         android:width="20dp"/>
7 </shape>
```

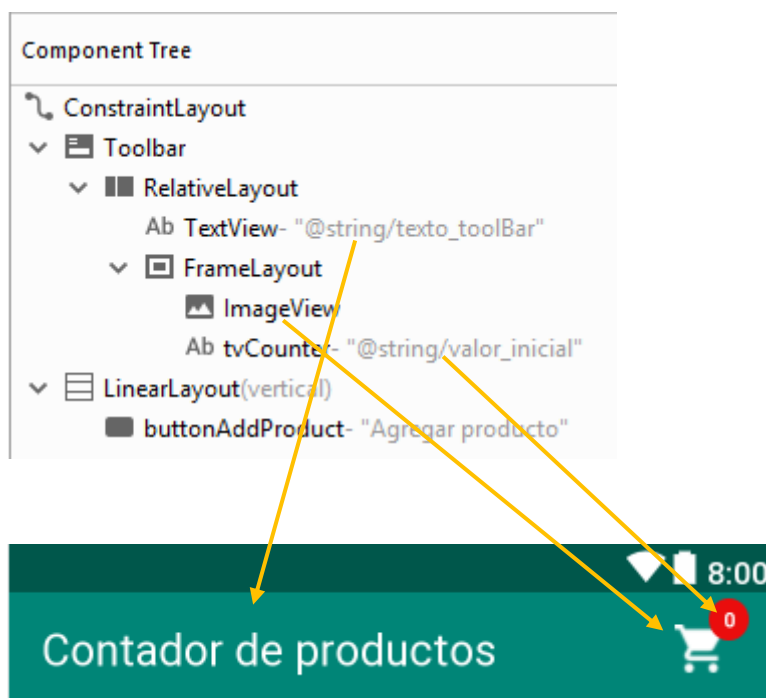
Definimos un círculo con su color y ancho y alto.

No podemos dedicar mucho tiempo, pues se sale de los cometidos del curso, a la definición de recursos drawables a través de documentos XML. Se recomienda que practiquéis por vuestra cuenta y realicéis algún que otro ejercicio.

## PASO2

Definimos el xml de la activity principal. En ella incluimos la Toolbar y el botón que permitirá incluir elementos en el carrito.

La definición de la Toolbar seguirá el siguiente esquema:



El TextView tvCounter, se define con un background que corresponde a la imagen definida por nosotros llamada rounded\_back.

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:app="http://schemas.android.com/apk/res-auto"
4      xmlns:tools="http://schemas.android.com/tools"
5      android:layout_width="match_parent"
6      android:layout_height="match_parent"
7      tools:context=".MainActivity">
8
9      <android.support.v7.widget.Toolbar
10         android:layout_width="match_parent"
11         android:layout_height="wrap_content"
12         android:background="@color/colorPrimary">
13
14         <RelativeLayout
15             android:layout_width="match_parent"
16             android:layout_height="match_parent">
17
18             <TextView
19                 android:layout_width="wrap_content"
20                 android:layout_height="wrap_content"
21                 android:text="@string/texto_toolBar"
22                 android:textColor="@android:color/white"
23                 android:textSize="22sp"
24                 android:layout_centerVertical="true"
25                 android:layout_alignParentLeft="true"
26                 android:layout_alignParentStart="true"/>
27
28             <FrameLayout
29                 android:layout_width="50dp"
30                 android:layout_height="50dp"
31                 android:layout_marginRight="15dp"
32                 android:layout_alignParentTop="true"
33                 android:layout_alignParentRight="true"
34                 android:layout_alignParentEnd="true">
35
36                 <ImageView
37                     android:layout_width="40dp"
38                     android:layout_height="40dp"
39                     android:padding="5dp"
40                     android:src="@drawable/carro_compra"
41                     android:scaleType="centerInside"
42                     android:layout_gravity="center"/>
43
44                 <TextView
45                     android:id="@+id/tvCounter"
46                     android:layout_width="wrap_content"
47                     android:layout_height="wrap_content"
48                     android:paddingLeft="5dp"
49                     android:paddingRight="5dp"
50                     android:paddingTop="4dp"
51                     android:paddingBottom="4dp"
52                     android:background="@drawable/rounded_back"
53                     android:text="@string/valor_inicial"
54                     android:gravity="center"
55                     android:textStyle="bold"
56                     android:textColor="@android:color/white"
57                     android:textSize="10sp"
58                     android:layout_gravity="right|top"/>
59             </FrameLayout>
60         </RelativeLayout>
61     </android.support.v7.widget.Toolbar>
62
63     <LinearLayout
64         android:layout_width="match_parent"
65         android:layout_height="match_parent"
66         android:orientation="vertical"
67         android:gravity="center">
68
69         <Button
70             android:id="@+id/buttonAddProduct"
71             android:layout_width="wrap_content"
72             android:layout_height="wrap_content"
73             android:text="Agregar producto"/>
74     </LinearLayout>
75 </android.support.constraint.ConstraintLayout>

```

## PASO3

Veamos el código correspondiente a la clase .java:

```
activity_main.xml × strings.xml × MainActivity.java ×
1 package com.example.pepe.profesor.contadorcompra;
2
3 import ...
4
5
6
7
8
9
10 public class MainActivity extends AppCompatActivity {
11
12     private Toolbar toolbar;
13     private TextView tvCounter;
14     private Button buttonAddProduct;
15
16     @Override
17     protected void onCreate(Bundle savedInstanceState) {
18         super.onCreate(savedInstanceState);
19         setContentView(R.layout.activity_main);
20
21         setContentView(R.layout.activity_main);
22         tvCounter = (TextView) findViewById(R.id.tvCounter);
23         buttonAddProduct = (Button) findViewById(R.id.buttonAddProduct);
24
25         buttonAddProduct.setOnClickListener(new View.OnClickListener() {
26             @Override
27             public void onClick(View v) {
28                 int counter = Integer.parseInt(tvCounter.getText().toString());
29                 ++counter;
30                 tvCounter.setText(String.valueOf(counter));
31             }
32         });
33     }
34 }
```